What is claimed is:

1. A computer-implemented method of debugging an object-oriented computer program, the method comprising:

(a) in response to user input, identifying a plurality of creators for a class defined in the object-oriented computer program and setting a plurality of breakpoints on the identified creators; and

(b) halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints.

2. The method of claim 1, wherein identifying the plurality of creators includes identifying every creator for the class.

3. The method of claim 1, further comprising, after identifying the plurality of creators, displaying a list of the identified creators and receiving user input to select a subset of identified creators, wherein the plurality of breakpoints are set on only the subset of the identified creators.

4. The method of claim 1, wherein the plurality of breakpoints are collectively set on all of the identified creators in response to the user input.

5. The method of claim 1, wherein setting the plurality of breakpoints includes setting each breakpoint from the plurality of breakpoints on a statement in one of the identified creators.

6. The method of claim 5, wherein setting each breakpoint includes inserting debugging program code in the creator on which such breakpoint is set.

7. The method of claim 1, wherein identifying the plurality of creators and setting the plurality of breakpoints are performed in response to user input to set a creation breakpoint, and wherein the plurality of breakpoints are associated with the creation breakpoint.

8. The method of claim 7, further comprising, in response to the user input to set the creation breakpoint, adding an entry for the creation breakpoint in a breakpoint data structure, wherein setting the plurality of breakpoints includes storing breakpoint information for each breakpoint in the breakpoint data structure, wherein the breakpoint information for each breakpoint is associated with the entry in the breakpoint data structure for the creation breakpoint.

9. The method of claim 1, further comprising tracking a total number of hits to the plurality of breakpoints.

10. The method of claim 9, wherein halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints includes:

(a) determining whether the total number of hits meets a condition in response to hitting any of the plurality of breakpoints; and

(b) halting execution of the object-oriented computer program if the total number of hits meets the condition.

11. The method of claim 10, wherein the condition is the total number of hits meeting or exceeding a threshold.

12. The method of claim 1, wherein each creator comprises a constructor method defined in the class.

13. The method of claim 1, further comprising collectively removing the plurality of breakpoints in response to user input.

14. A computer-implemented method of debugging an object-oriented computer program, the method comprising:

(a) tracking a number of object creations of a class defined in the object-oriented computer program during debugging; and

(b) halting execution of the object-oriented computer program in response to the number of object creations meeting a condition.

15. The method of claim 14, wherein the condition is the number of object creations meeting or exceeding a threshold.

16. The method of claim 14, wherein tracking the number of object creations includes incrementing a counter in response to hitting any of a plurality of breakpoints set on a plurality of creators for the class.

17. The method of claim 14, further comprising, in response to user input, identifying the plurality of creators for the class and setting the plurality of breakpoints on the identified creators.

18. The method of claim 17, wherein identifying the plurality of creators includes identifying every creator for the class.

19. The method of claim 17, further comprising, after identifying the plurality of creators, displaying a list of the identified creators and receiving user input to select a subset of identified creators, wherein the plurality of breakpoints are set on only the subset of the identified creators.

20. The method of claim 17, wherein the plurality of breakpoints are collectively set on all of the identified creators in response to the user input.

21. The method of claim 17, wherein identifying the plurality of creators and setting the plurality of breakpoints are performed in response to user input to set a

creation breakpoint, and wherein the plurality of breakpoints are associated with the creation breakpoint.

22. The method of claim 18, wherein each creator comprises a constructor method defined in the class.

23. An apparatus, comprising:

(a) a memory within which resides at least a portion of an object-oriented computer program; and

(b) program code configured to debug the object-oriented computer program by, in response to user input, identifying a plurality of creators for a class defined in the object-oriented computer program and setting a plurality of breakpoints on the identified creators, and halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints.

24. The apparatus of claim 23, wherein the program code is configured to identify the plurality of creators by identifying every creator for the class.

25. The apparatus of claim 23, wherein the program code is further configured to, after identifying the plurality of creators, display a list of the identified creators and receive user input to select a subset of identified creators, wherein the plurality of breakpoints are set on only the subset of the identified creators.

26. The apparatus of claim 23, wherein the plurality of breakpoints are collectively set on all of the identified creators in response to the user input.

27. The apparatus of claim 23, wherein the program code is configured to set the plurality of breakpoints by setting each breakpoint from the plurality of breakpoints on a statement in one of the identified creators.

28. The apparatus of claim 23, wherein the program code is configured to identify the plurality of creators and set the plurality of breakpoints in response to user input to set a creation breakpoint, and wherein the plurality of breakpoints are associated with the creation breakpoint.

29. The apparatus of claim 28, further comprising a breakpoint data structure, resident in the memory, wherein the program code is configured to, in response to the user input to set the creation breakpoint, add an entry for the creation breakpoint in the breakpoint data structure, and wherein the program code is configured to set the plurality of breakpoints by storing breakpoint information for each breakpoint in the breakpoint data structure, wherein the breakpoint information for each breakpoint is associated with the entry in the breakpoint data structure for the creation breakpoint.

30. The apparatus of claim 23, wherein the program code is further configured to track a total number of hits to the plurality of breakpoints.

31. The apparatus of claim 30, wherein the program code is configured to determine whether the total number of hits meets a condition in response to hitting any of the plurality of breakpoints, and halt execution of the object-oriented computer program if the total number of hits meets the condition.

32. The apparatus of claim 31, wherein the condition is the total number of hits meeting or exceeding a threshold.

33. The apparatus of claim 23, wherein the program code is further configured to collectively remove the plurality of breakpoints in response to user input.

34. An apparatus, comprising:

(a) a memory within which resides at least a portion of an object-oriented computer program; and

(b) program code configured to debug the object-oriented computer program by tracking a number of object creations of a class defined in the object-oriented computer program during debugging, and halting execution of the object-oriented computer program in response to the number of object creations meeting a condition.

35. The apparatus of claim 34, wherein the condition is the number of object creations meeting or exceeding a threshold.

36. The apparatus of claim 34, wherein the program code is configured to track the number of object creations by incrementing a counter in response to hitting any of a plurality of breakpoints set on a plurality of creators for the class, and wherein the program code is further configured to, in response to user input, identify the plurality of creators for the class and set the plurality of breakpoints on the identified creators.

37. The apparatus of claim 36, wherein the program code is configured to identify the plurality of creators and set the plurality of breakpoints in response to user input to set a creation breakpoint, and wherein the plurality of breakpoints are associated with the creation breakpoint.

38. A program product, comprising:

(a) program code configured to debug an object-oriented computer program by, in response to user input, identifying a plurality of creators for a class defined in the object-oriented computer program and setting a plurality of breakpoints on the identified creators, and halting execution of the object-oriented computer program during debugging in response to hitting any of the plurality of breakpoints; and

(b) a signal bearing medium bearing the program code.

39. The program product of claim 38, wherein the signal bearing medium includes at least one of a transmission medium and a recordable medium.

40. A program product, comprising:

(a) program code configured to debug an object-oriented computer program by tracking a number of object creations of a class defined in the object-oriented computer program during debugging, and halting execution of the object-oriented computer program in response to the number of object creations meeting a condition; and

(b) a signal bearing medium bearing the program code.